



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,107	05/04/2001	Stephen Andrew Brodsky	STL920000075US2	1986
24852	7590	10/06/2004	EXAMINER	
INTERNATIONAL BUSINESS MACHINES CORP			BAYARD, DJENANE M	
IP LAW			ART UNIT	PAPER NUMBER
555 BAILEY AVENUE , J46/G4			2141	
SAN JOSE, CA 95141				

DATE MAILED: 10/06/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/849,107	BRODSKY ET AL.
	Examiner Djenane M Bayard	Art Unit 2141

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
 - If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
 - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
 - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 04 May 2001.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-67 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-67 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>2/12/02</u> . | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Double Patenting

1. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

2. Claims 1-4, 16-19, 30-33, 44-45, 55-60 are provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-4, 6-9, 11-14, 16-17, 19-24 of copending Application No. 09/849377. This is a provisional double patenting rejection since the conflicting claims have not in fact been patented.

Claim Objections

3. Claims 1, 3 and 15 are objected to because of the following informalities:

Claim 1 cites "theresponse" in step d, it should be "the response".

Claim 3 cites "an connector", it should be "a connector".

Claim 15 cites "sourcelanguage", it should be "source language".

Appropriate correction is required.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 55-67 are rejected under 35 U.S.C. 101 because the disclosed invention is inoperative and therefore the claimed invention lacks patentable utility. Claims 55 claims a product only without any function, use and purpose.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1-4, 6-19, 21-25, 27-33, 35-45, 47-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application No. 2003/0191970 to Devine et al in view of U.S. Patent No. 6,738,975 to Yee et al.

- a. As per claim 1, Devine et al teaches a) initiating the application request on the end user application in a first language with a first application program (See page 4, paragraph 0058 and page 5, paragraph [0068]); b) transmitting the application request to the server (See page 5,

paragraph [0068]), and converting the application request from the first language of the first end user application to a language running on the application server (See page 8, paragraph [0099])

Remark : Devine teaches converting the application request from the first language of the first end user application to the legacy language application running on the application server); c) processing said application request on the application server (See page 5, paragraph [0073]); d) transmitting a response to the application request from the application server to the end user application and converting the response to the application request from the language running on the application server to the first language of the first end user application (See page 8, paragraph [0099])and e) wherein the end user application and the application server have at least one connector therebetween (See page 3, paragraph [0054] and figure 1) and the steps of (i) converting the application request from the first language of the first end user application as a source language to the language running on the application server as a target language (See page 8, paragraph [0099]), and (ii) converting a response to the application request from the language running on the application server as a source language to the first language of the first end user application as a target language (See page 6, paragraph [0074] and page 8 paragraph [0099]) and converting the source language to the target language (See page 6, paragraph [0074]). However, Devine fails to teach invoking connector metamodels of respective source and target languages; 2) populating the connector metamodels with metamodel data of each of the respective source and target languages.

Yee et al teaches an extensible distributed enterprise application integration system.

Furthermore, Yee et al teaches invoking connector metamodels of respective source and target

languages; 2) populating the connector metamodels with metamodel data of each of the respective source and target languages (See col. 21, lines 1-11 and col. 21, lines 43-55)

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate invoking connector metamodels of respective source and target languages; 2) populating the connector metamodels with metamodel data of each of the respective source and target languages taught by Yee et al in order to reduce or substantially eliminate the need for the expensive custom coding that is traditionally required to integrate applications (See col. 11, lines 15-19).

b. As per claims 16, 30 and 44, Devine et al teaches a) the client having an end user application, and being controlled and configured to initiate an application request with the server in a first language with a first application program and to transmit the application request to the server (See page 4, paragraph [0058] and page 6 paragraph [0074]); b) the connector being configured and controlled to receive the application request from the client, convert the application request from the first language of the first end user application running on the client to a language running on the server (See page 5, paragraph [00668] and page 7, paragraph[0095] ; c) the server being configured and controlled to receive the converted application request from the connector and processing the said application request in a second language with a second application program residing on the server, and to thereafter transmit a response to the application request through the connector back to the first application program on the client (See page 5, paragraph [0068]); d) the connector being configured and controlled to receive a response to the application request from the server, to convert a response to the application

request from the language running on the application server to the first language of the first application program running on the client; and e) wherein connector between the client and the server is configured and controlled to (i) convert the application request from the first language of the client application on the client as a source language to the language running on the application server as a target language (See page 8, paragraph [0099]), and (ii) convert the response to the application request from the language running on the application server as a source language to the first language of the client application running on the client as a target language (See page 4, paragraph [0058]). However, Devine et al fails to teach 1) retrieving connector metamodels of respective source and target languages from a metamodel metadata repository; 2) populating the connector metamodels with metamodel data from the metamodel metadata repository for each of the respective source and target languages; and 3) invoking the retrieved, populated connector metamodels and converting the source language to the target language.

Yee et al teaches an extensible distributed enterprise application integration system. Furthermore, Yee et al teaches 1) retrieving connector metamodels of respective source and target languages from a metamodel metadata repository; 2) populating the connector metamodels with metamodel data from the metamodel metadata repository for each of the respective source and target languages; and 3) invoking the retrieved, populated connector metamodels and converting the source language to the target language (See col. 21, lines 1-11 and col. 21, lines

43-55

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate retrieving connector metamodels of respective source and target

languages from a metamodel metadata repository; 2) populating the connector metamodels with metamodel data from the metamodel metadata repository for each of the respective source and target languages; and 3) invoking the retrieved, populated connector metamodels and converting the source language to the target language as taught by Yee et al in the claimed invention of Devine et al in order to reduce or substantially eliminate the need for the expensive custom coding that is traditionally required to integrate applications (See col. 11, lines 15-19).

c. As per claims 2, 17 and 31, Devine et al teaches wherein the end user application is a web browser (See page 3, paragraph [0056]).

d. As per claims 3, 18, and 32, Devine et al teaches wherein the end user application is connected to the application server through a web server, and the web server comprises a connector (See page 3, paragraph [0053-0055]).

e. As per claims 4, 19, 33 and 45, Devine et al teaches the claimed invention as described above. However, Devine et al fails to teach wherein the metamodel metadata comprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata.

Yee et al teaches an extensible distributed enterprise application integration system. Furthermore, Yee et al teaches an agent adapter mediate differences interface protocols and data structures between applications to provide a uniform, normalized view of business event (See col. 16, lines 62-65), message definition object identifies data of an enterprise application, and

well-defined message format to describe the layout of the native data (See col. 17, lines 49-54 and 48-61), and mapping definition objects define how the system will transform system message extract from one or more applications to message needed by other applications (See col. 17, lines 55-58 and col. 20, lines 53-59).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate wherein the metamodel metadata comprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata as taught by Yee et al in the claimed invention of Devine et al in order to reduce or substantially eliminate the need for the expensive custom coding that is traditionally required to integrate applications (See col. 11, lines 15-19).

f. As per claims 6,21, 35 and 47, Devine et al fails to teach wherein the application domain interface metamodel metadata comprises input parameter signatures, output parameter signatures, and return types.

Yee et al teaches an agent adapter that mediates differences interface protocols and data structures between application (See col. 16, lines 62-65), and user can create a transformation to transform input data from the source application to the input format expected by the target application (See col. 21, lines 12-24).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate wherein the application domain interface metamodel metadata comprises input parameter signatures, output parameter signatures, and return types as taught by Yee et al in the claimed invention of Devine et al in order to provide a system for integrating a

plurality of computer applications (See col. 11, lines 65-67)

g. As per claims 7,22, 36 and 48, Devine et al fails to teach wherein the application domain interface metamodel metadata further includes language metamodel metadata.

Yee et al teaches language metamodel metadata (See col. 18, lines 48-61).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate wherein the application domain interface metamodel metadata further includes language metamodel metadata as taught by Yee et al in the claimed invention of Devine et al in order to provide a system for integrating a plurality of computer applications (See col. 11, lines 65-67).

h. As per claims 8,23,37 and 49, Devine et al fails to teach wherein the language metamodel metadata includes mappings between source and target languages.

Yee et al teaches teach wherein the language metamodel metadata includes mappings between source and target languages (See col. 18, lines 48-61).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate teach wherein the language metamodel metadata includes mappings between source and target languages as taught by Yee et al in the claimed invention of Devine et al in order to integrate a plurality of computer applications (See col. 11, lines 65-67).

i. As per claims 9,24, 38 and 50, Devine et al teaches wherein the source language is object oriented and wherein the target is not object oriented (See page 4, paragraph [0056]). Devine et

al fails to teach wherein the language metamodel metadata maps encapsulated objects into code and data. However, it is inherent to one with ordinary skill in the art at the time the invention was made that any object oriented language could pass object as argument in the request, and the system of Devine could translate the request into message of the format that is needed by the Legacy application.

j. As per claims 10, 25, 40, and 52, Devine fails to teach wherein the language metamodel metadata maps object inheritances into references and pointers. However, it is well known in the art that object oriented class can inherit in parent class. It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate wherein the language metamodel metadata maps object inheritances into references and pointers because in order map from one language to another language, reference to parent class must also be mapped.

k. As per claims 11, 26, 39 and 51, Devine et al fails to teach wherein the source language and the target language are different object oriented languages, and the language metamodel metadata maps encapsulated code and data between the languages.

Yee et al teaches wherein the source language and the target language are different object oriented languages, and the language metamodel metadata maps encapsulated code and data between the languages (See col. 18, lines 48-61).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate wherein the source language and the target language are different object oriented languages, and the language metamodel metadata maps encapsulated code and data

between the languages as taught by Yee et al in the claimed invention of Devine et al in order to integrate a plurality of computer applications (See col. 11, lines 65-67).

- l. As per claims 12, 27, 41 and 53, Devine et al teaches wherein the type descriptor metamodel metadata defines physical realizations, storage mapping, data types, data structures, and realization constraints (See page 3, paragraph [0053-0055]).
- m. As per claim 13, Devine et al teaches wherein the transaction is a rich transaction comprising a plurality of individual transactions, and further comprising processing the plurality of individual transactions on one end user application and a plurality of application servers (See page 3, paragraph [0053-0055]).
- n. As per claims 14, Devine et al teaches passing individual transactions among individual application servers (See page 3, paragraph [0053-0055]).
- o. As per claims 15, 29, 43 and 54, Devine et al fails to teach wherein one of the source language or the target language is chosen from the group consisting of C and C++.
Yee et al teaches wherein one of the source language or the target language is chosen from the group consisting of C and C++.

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate one of the source language or the target language is chosen from the

group consisting of C and C++ as taught by Yee et al in the claimed invention of Devine et al in order to integrate a plurality of computer applications (See col. 11, lines 65-67).

p. As per claims 28 and 42, Devine et al teaches wherein the system has a plurality of applications servers and is configured and controlled to process rich transaction (See page 4, paragraph [0061-0065]).

8. Claims 5, 20,34 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application No. 2003/0191970 to Devine et al in view of U.S. Patent No. 6,738,975 to Yee et al. as applied to claim 4 above, and further in view of U.S. Patent No. 6,094,688 to Mellen-Garnett et al.

a. As per claims 5,20,34 and 46, Devine wherein the invocation metamodel metadata is chosen from the group consisting of message control information, security data and user data (See page 5, paragraph [0068] and page 7, paragraph [0095]). However, Devine fails to teach transactional semantics, trace and debug information, pre-condition and post-condition resources.

Mellen-Garnett teaches transactional semantics (See col. 7, lines 46-62), trace and debug information (Error and exception service 260, See col. 10, lines 23-40), pre-condition and post-condition resources (transaction system resources, See col. 57-67).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate transactional semantics, trace and debug information, pre-condition and post-condition resources as taught by Mellen-Garnett in the claimed invention of Devine et al in

view of Yee et al in order to provide a modular application collaborator for providing inter-operability between application (See col. 1, lines 51-55).

9. Claims 55-60 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,738,975 to Yee et al in view of U.S. Patent No. 6,094,688 Mellen-Garnett et al.

a. As per claim 55, Yee et al teaches a program product comprising a storage medium having application domain interface metamodel metadata lines and language metamodel metadata, and computer instructions for building a metamodel metadata repository of source and target language metamodel metadata (See col. Col. 16, lines 62-65, col. 17, lines 49-54 and col. 20, lines 53-59). However, Yee et al fails to teach invocation metamodel metadata.

Mellen-Garnett et al teaches invocation metamodel metadata (See col. 19, lines 6-10).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate invocation metamodel metadata as taught by Mellen-Garnet et al in the claimed invention of Yee et al in order to provide inter-operability between applications (See col. 1, lines 52-55).

b. As per claim 56, Yee et al teaches computer instructions for building connector stubs from said metamodel metadata (See col. 26, lines 47-62 and col. 15, lines 41-54).

c. As per claim 57, Yee et al teaches 1) retrieving connector metamodel metadata of respective source and target languages from the metamodel metadata repository (See col. 15,

lines 55-58); 2) populating the connector metamodels with metamodel data of each of the respective source and target languages from the metamodel metadata repository and invoking the retrieved, populated connector metamodels (See col. 21, lines 49-55); and 3) converting the source language to the target language (See col. 21, lines 49-65 and col. 22, line 6).

d. As per claim 58, Yee et al teaches wherein the metamodel metadata in the repository comprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata (See col. 16, lines 62-65 and col. 17, lines 48-61).

e. As per claim 59, Yee et al fails to teaches wherein the invocation metamodel metadata is chosen from the group consisting of message control information, security data, transactional semantics, trace and debug information, pre-condition and post-condition resources, and user data.

Mellen-Garnett teaches wherein the invocation metamodel metadata is chosen from the group consisting of message control information, security data, transactional semantics, trace and debug information, pre-condition and post-condition resources, and user data.

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate wherein the invocation metamodel metadata is chosen from the group consisting of message control information, security data, transactional semantics, trace and debug information, pre-condition and post-condition resources, and user data as taught by Mellen-Garnett in the claimed invention of Yee et al in order to provide inter-operability

between applications (See col. 1, lines 52-55).

f. As per claim 60, Yee et al teaches wherein the application domain interface metamodel metadata comprises input parameter signatures, output parameter signatures, and return types (See col. 16, lines 62-65 and col. 21, lines 12-24).

10. Claims 61-67 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,738,975 to Yee et al in view of U.S. Patent No. 6,094,688 Mellen-Garnett et al as applied to claim 55 above, and further in view of U.S. Patent Application No. 2003/0191970 to Devine et al.

a. As per claim 61, Yee et al fails teaches wherein the application domain interface metamodel metadata further includes language metamodel metadata (See col. 18, lines 48-61).

b. As per claim 62, Yee et al wherein the language metamodel metadata includes mappings between source and target languages (See col. 18, lines 48-61).

c. As per claim 63, Yee et al teaches wherein one of the source or target languages is object oriented (See col. 5, lines 50-55, Yee teaches wherein the language of the user application can be JAVA). Yee et al fails to teach wherein the language metamodel metadata maps encapsulated objects into code and data. However, it is inherent to one with ordinary skill in the art at the time

the invention was made that any object oriented language could pass object as argument in the request, and the system of Yee could translate the request into message of the format that is needed by the Legacy application.

- d. As per claim 64, Yee et al teaches wherein the source language and the target language are different object oriented languages, and the language metamodel metadata maps encapsulated code and data between the languages (See col. 5, lines 50-55 and See col. 18, lines 48-61).
- e. The program product of claim 64 wherein the language metamodel metadata maps object inheritances into references and pointers.
- f. As per claim 66, Yee et al teaches wherein the type descriptor metamodel metadata defines physical realizations, storage mapping, data types, data structures, and realization constraints (See col. 18, lines 48-61).
- g. As per claim 67, Yee et al teaches wherein one of the source and target languages is chosen from the group consisting of C and C++ (See col. 27, lines 55-65).

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent No. 6,711,734 to Baisley teaches a method for translating MOF metamodels to UML Models

U.S. Patent No. 6,134,559 to Brumme et al teaches an uniform object model having methods and additional features for integrating objects defined by different foreign object type systems into a single type system.

U.S. Patent Application No. 2002/0112078 to Yach teaches a virtual machine web browser.

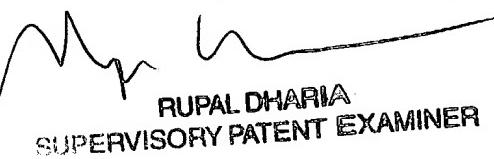
U.S. Patent Application No. 6,480,860 to Monday teaches a tagged markup language interface with document type definition to access data in object-oriented database.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Djenane M Bayard whose telephone number is (703) 305-6606. The examiner can normally be reached on Monday- Friday 5:30 AM- 3:00 PM..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rupal Dharia can be reached on (703) 305-4003. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Djenane Bayard



RUPAL DHARIA
SUPERVISORY PATENT EXAMINER